

AAAAAA
AAAAAA
AA AA
AA AA
AA AA
AA AA
AA AA
AAAAAAAA
AAAAAAAA
AA AA
AA AA
AA AA
AA AA

EEEEEEEE
EEEEEEEE
EE
EE
EE
EE
EEEEEEEE
EEEEEEEE
EE
EE
EE
EEEEEEEE
EEEEEEEE

DDDDDDDD
DDDDDDDD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DDDDDDDD
DDDDDDDD

DDDDDDDD
DDDDDDDD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DDDDDDDD
DDDDDDDD

EEEEEEEE
EEEEEEEE
EE
EE
EE
EE
EEEEEEEE
EEEEEEEE
EE
EE
EE
EEEEEEEE
EEEEEEEE

CCCCCCCC
CCCCCCCC
CC
CC
CC
CC
CC
CC
CC
CC
CC
CCCCCCCC
CCCCCCCC

000000
000000
00
00
00
00
00
00
00
00
00
000000
000000

DDDDDDDD
DDDDDDDD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DD DD
DDDDDDDD
DDDDDDDD

EEEEEEEE
EEEEEEEE
EE
EE
EE
EE
EEEEEEEE
EEEEEEEE
EE
EE
EE
EEEEEEEE
EEEEEEEE

....
....
....
....

LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LLLLLLLL
LLLLLLLL

IIIIII
IIIIII
II
II
II
II
II
II
II
II
II
II
IIIIII
IIIIII

SSSSSSSS
SSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSS
SSSSSSSS

```
0001 0 MODULE AED$DECODE (
0002 0     LANGUAGE (BLISS32),
0003 0     IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 *   ALL RIGHTS RESERVED.
0012 1 *
0013 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 *   TRANSFERRED.
0019 1 *
0020 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 *   CORPORATION.
0023 1 *
0024 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *****
0028 1
0029 1 ++
0030 1
0031 1 FACILITY:      Miscellaneous utilities
0032 1
0033 1 ABSTRACT:
0034 1
0035 1     This module contains the routines necessary to read the action
0036 1     definition file and decode the users input based upon the action
0037 1     definitions.
0038 1
0039 1 ENVIRONMENT:
0040 1
0041 1     VAX/VMS operating system, user mode utilities.
0042 1
0043 1 --
0044 1
0045 1
0046 1
0047 1 AUTHOR:        L. Mark Pilant      CREATION DATE: 15-Sep-1982 15:30
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1     V03-005 LMP0213      L. Mark Pilant,      24-Mar-1984 12:23
0052 1     Add support for locking and unlocking the object's ACL.
0053 1
0054 1     V03-004 LMP0193      L. Mark Pilant,      14-Feb-1984 10:04
0055 1     Add support for additional edition actions: delete BDL,
0056 1     session reset, and quit session.
0057 1
```



```
: 58      0058 1 |
: 59      0059 1 |
: 60      0060 1 |
: 61      0061 1 |
: 62      0062 1 |
: 63      0063 1 |
: 64      0064 1 |
: 65      0065 1 |
: 66      0066 1 |
: 67      0067 1 |
: 68      0068 1 |
: 69      0069 1 |
: 70      0070 1 |
: 71      0071 1 |
: 72      0072 1 |

V03-003 LMP0172      L. Mark Pilant,      28-Nov-1983 12:11
        Numerous bug fixes, support for VT2xx terminals, and a
        session keystroke logger.

V03-002 LMP0142      L. Mark Pilant,      24-Aug-1983 3:17
        Change references to ACLEDIT$INI to be ACLEDIT$INIT.

V03-001 LMP0103      L. Mark Pilant,      21-Apr-1983 12:44
        Add support for HIDDEN and PROTECTED ACES.

        **

LIBRARY 'SYSS$LIBRARY:LIB.L32';
LIBRARY 'SYSS$LIBRARY:TPAMAC.L32';
REQUIRE 'SRC$ACLEDTDEF';
```

```
74 0525 1 FORWARD ROUTINE
75 0526 1 AED_GETKEYINI,          ! Check for & read definition file
76 0527 1 AED_DECODEKEY,        ! Decode input given definitions
77 0528 1 AED_FLUSHKEY,         ! Flush session buffer & close file
78 0529 1
79 0530 1 ! TPARSE action routine.
80 0531 1
81 0532 1 SET RUBOUT,           ! Set rubout as the string definition
82 0533 1 SET_DEFINITION;       ! Define a key
83 0534 1
84 0535 1 EXTERNAL ROUTINE
85 0536 1 AED_FILEERROR : NOVALUE, ! RMS file error reporting
86 0537 1 AED_PUTOUTPUT,       ! General purpose output routine
87 0538 1 AED_SET_CURSOR;      ! Set cursor position & remember
88 0539 1
89 0540 1 EXTERNAL
90 0541 1 KEY_TABLE : $BBLOCK [8]; ! Key definition table listhead
91 0542 1
92 0543 1 ! Storage for TPARSE usage.
93 0544 1
94 0545 1 OWN
95 0546 1 KEY_BLOCK : $BBLOCK [KEY_C_LENGTH], ! Key definition block
96 0547 1 KEY_STRING : $BBLOCK [DSC$C_S_BLN]; ! Key string descriptor
97 0548 1
98 0549 1 BIND
99 0550 1 KEY_ACTION = KEY_BLOCK[KEY_B_ACTION] : BYTE, ! Action code
100 0551 1 KEY_FLAGS = KEY_BLOCK[KEY_B_FLAGS] : BYTE; ! Needed flags
101 0552 1
102 0553 1 ! TPARSE state tables to parse the action definition file.
103 0554 1
104 0555 1 $INIT_STATE (KEYDEF_STATE, KEYDEF_KEY);
105 0556 1
106 P 0557 1 $STATE (SWALLOW 1,
107 P 0558 1 (TPAS BLANK, SWALLOW_1),
108 P 0559 1 ('DEFINE')
109 0560 1 );
110 0561 1
111 P 0562 1 $STATE (SWALLOW 2,
112 P 0563 1 (TPAS BLANK, SWALLOW 2),
113 P 0564 1 ('GOLD'...KEY_C_GOLD, KEY_ACTION),
114 P 0565 1 ('HELP'...KEY_C_HELP, KEY_ACTION),
115 P 0566 1 ('HELP FORMAT'...KEY_C_HELPFMT, KEY_ACTION),
116 P 0567 1 ('LOCATE STRING'...KEY_C_FIND_STR, KEY_ACTION),
117 P 0568 1 ('LOCATE NEXT'...KEY_C_FIND_NXT, KEY_ACTION),
118 P 0569 1 ('DELETE ACE'...KEY_C_DEL_ACE, KEY_ACTION),
119 P 0570 1 ('UNDELETE ACE'...KEY_C_UNDEL_ACE, KEY_ACTION),
120 P 0571 1 ('SELECT FIELD'...KEY_C_SEL_FIELD, KEY_ACTION),
121 P 0572 1 ('ADVANCE FIELD'...KEY_C_ADV_FIELD, KEY_ACTION),
122 P 0573 1 ('DELETE WORD'...KEY_C_DEL_WRD, KEY_ACTION),
123 P 0574 1 ('UNDELETE WORD'...KEY_C_UNDEL_WRD, KEY_ACTION),
124 P 0575 1 ('ADVANCE POSITION'...KEY_C_ADVANCE, KEY_ACTION),
125 P 0576 1 ('BACKUP POSITION'...KEY_C_BACKUP, KEY_ACTION),
126 P 0577 1 ('DELETE CHARACTER'...KEY_C_DEL_CHR, KEY_ACTION),
127 P 0578 1 ('UNDELETE CHARACTER'...KEY_C_UNDEL_CHR, KEY_ACTION),
128 P 0579 1 ('MOVE WORD'...KEY_C_MOVE_WRD, KEY_ACTION),
129 P 0580 1 ('MOVE ACE'...KEY_C_MOVE_ACE, KEY_ACTION),
130 P 0581 1 ('MOVE_EOL'...KEY_C_MOVE_EOL, KEY_ACTION),
```



```
131 P 0582 1 ('DELETE_EOL',...KEY_C_DEL_EOL,KEY_ACTION),
132 P 0583 1 ('INSERT_ACE',...KEY_C_INSERT,KEY_ACTION),
133 P 0584 1 ('SELECT_ITEM',...KEY_C_SEL_ITEM,KEY_ACTION),
134 P 0585 1 ('ENTER_ACE',...KEY_C_ENTER,KEY_ACTION),
135 P 0586 1 ('PREVIOUS_SCREEN',...KEY_C_PREV_SCREEN,KEY_ACTION),
136 P 0587 1 ('NEXT_SCREEN',...KEY_C_NEXT_SCREEN,KEY_ACTION),
137 P 0588 1 ('UP_ARROW',...KEY_C_UP,KEY_ACTION),
138 P 0589 1 ('DOWN_ARROW',...KEY_C_DOWN,KEY_ACTION),
139 P 0590 1 ('RIGHT_ARROW',...KEY_C_RIGHT,KEY_ACTION),
140 P 0591 1 ('LEFT_ARROW',...KEY_C_LEFT,KEY_ACTION),
141 P 0592 1 ('INSERT_OVERSTRIKE',...KEY_C_OVERSTRIKE,KEY_ACTION),
142 P 0593 1 ('MOVE_BOL',...KEY_C_MOVE_BOL,KEY_ACTION),
143 P 0594 1 ('RUBOUT_WORD',...KEY_C_RUB_WRD,KEY_ACTION),
144 P 0595 1 ('SCREEN_REFRESH',...KEY_C_REFRESH,KEY_ACTION),
145 P 0596 1 ('SESSION_RESET',...KEY_C_RESET,KEY_ACTION),
146 P 0597 1 ('RUBOUT_BOL',...KEY_C_RUB_BOL,KEY_ACTION),
147 P 0598 1 ('UNDELETE_LINE',...KEY_C_UNDEL_LIN,KEY_ACTION),
148 P 0599 1 ('EXIT',...KEY_C_EXIT,KEY_ACTION),
149 P 0600 1 ('QUIT_SESSION',...KEY_C_QUIT,KEY_ACTION),
150 P 0601 1 ('RUBOUT_CHARACTER',...KEY_C_RUB_CHR,KEY_ACTION),
151 0602 1 );
152 0603 1
153 P 0604 1 $STATE (SWALLOW_3,
154 P 0605 1 (TPAS_BLANK,SWALLOW_3),
155 P 0606 1 ('AS')
156 0607 1 );
157 0608 1
158 P 0609 1 $STATE (KEY_DEFINE,
159 P 0610 1 (TPAS_BLANK,KEY_DEFINE),
160 P 0611 1 ('GOLD',...KEY_M_GOLDREQ,KEY_FLAGS),
161 P 0612 1 ('CONTROL',GET_TEXT,...KEY_M_CTRLCHAR,KEY_FLAGS),
162 P 0613 1 ('ESCAPE',GET_TEXT,...KEY_M_ESCSEQ,KEY_FLAGS),
163 P 0614 1 ('CSI',GET_TEXT,...KEY_M_CSI,KEY_FLAGS),
164 P 0615 1 ('SS3',GET_TEXT,...KEY_M_SS3,KEY_FLAGS),
165 P 0616 1 ('RUBOUT',SET_RUBOUT),
166 P 0617 1 (TPAS_EOS,TPAS_FAIL),
167 0618 1 );
168 P 0619 1 $STATE (CHECK_END,
169 P 0620 1 (TPAS_BLANK,CHECK_END),
170 P 0621 1 ('',KEY_DEFINE),
171 P 0622 1 ('OR',KEY_DEFINE,SET_DEFINITION),
172 P 0623 1 (TPAS_EOS,TPAS_EXIT,SET_DEFINITION)
173 0624 1 );
174 0625 1
175 P 0626 1 $STATE (GET_TEXT,
176 P 0627 1 (TPAS_BLANK,GET_TEXT),
177 P 0628 1 ('')
178 0629 1 );
179 P 0630 1 $STATE (SWALLOW_4,
180 P 0631 1 (TPAS_BLANK,SWALLOW_4),
181 P 0632 1 ((GET_STRING),CHECK_END,...KEY_STRING)
182 0633 1 );
183 0634 1
184 P 0635 1 $STATE (GET_STRING,
185 P 0636 1 ((CHECK_DELIM),GET_STRING),
186 P 0637 1 (TPAS_LAMBDA,TPAS_EXIT)
187 0638 1 );
```

AED\$DECODE
V04-000

F 16
15-Sep-1984 23:37:58
14-Sep-1984 11:52:23

VAX-11 Bliss-32 V4.0-742
[ACLEDT.SRC]AEDDECODE.B32;1

Page 5
(2)

```
: 188      P 0639 1 $STATE (CHECK DELIM  
: 189      P 0640 1      (',',TPAS_FAIL),  
: 190      P 0641 1      (',',TPAS_FAIL),  
: 191      P 0642 1      (TPAS_EOS,TPAS_FAIL),  
: 192      P 0643 1      (TPAS_ANY,TPAS_EXIT)  
: 193      0644 1      );
```

```
195 0645 1 GLOBAL ROUTINE AED_GETKEYINI =
196 0646 1
197 0647 1 ++
198 0648 1
199 0649 1 FUNCTIONAL DESCRIPTION:
200 0650 1
201 0651 1 This routine attempts to open the action definition file pointed
202 0652 1 to by the logical name ACLEDITSINIT. If the logical name does not
203 0653 1 exist a success return is given. If the logical name exists, but
204 0654 1 the file it points to does not, a warning message is given, and a
205 0655 1 success return is given. If any errors occur while reading the
206 0656 1 definition file, the appropriate error message is given.
207 0657 1
208 0658 1 CALLING SEQUENCE:
209 0659 1 AED_GETKEYINI ()
210 0660 1
211 0661 1 INPUT PARAMETERS:
212 0662 1 none
213 0663 1
214 0664 1 IMPLICIT INPUTS:
215 0665 1 none
216 0666 1
217 0667 1 OUTPUT PARAMETERS:
218 0668 1 none
219 0669 1
220 0670 1 IMPLICIT OUTPUTS:
221 0671 1 none
222 0672 1
223 0673 1 ROUTINE VALUE:
224 0674 1 1 if successful, logical name does not exist, or file does not exist
225 0675 1 error code otherwise
226 0676 1
227 0677 1 SIDE EFFECTS:
228 0678 1 none
229 0679 1
230 0680 1 --
231 0681 1
232 0682 1 BEGIN
233 0683 1
234 0684 1 LOCAL
235 0685 1 KEYINI_FAB : SFAB_DECL, ! Key definition file FAB
236 0686 1 KEYINI_RAB : SRAB_DECL, ! Key definition file RAB
237 0687 1 KEYINI_NAM : $NAM_DECL, ! Key definition file NAM block
238 0688 1 KEYINI_EXP_NAM : $BBLOCK [NAMSC_MAXRSS], ! Expanded name storage
239 0689 1 KEYINI_RES_NAM : $BBLOCK [NAMSC_MAXRSS], ! Resultant name storage
240 0690 1 DEFINE_LINE : VECTOR [512,BYTE], ! Line from definition file
241 0691 1 TPARSE_BLOCK : $BBLOCK [TPASK_LENGTH], ! Parser context block
242 0692 1 LINE_INDEX, ! Index into line read in
243 0693 1 LOCAL_STATUS; ! Local error status
244 0694 1
245 0695 1 ! Initialize the necessary RMS data structures.
246 0696 1
247 P 0697 1 $FAB_INIT (FAB = KEYINI_FAB,
248 P 0698 1 FAC = GET,
249 P 0699 1 FNA = UPLIT ('ACLEDITSINIT:'),
250 P 0700 1 FNS = %CHARCOUNT ('ACLEDITSINIT:'),
251 P 0701 1 FOP = SQO,
```



```
252 P 0702 NAM = KEYINI_NAM,
253 P 0703 ORG = SEQ,
254 P 0704 RFM = VAR);
255 P 0705 $NAM_INIT (NAM = KEYINI_NAM,
256 P 0706 ESA = KEYINI_EXP_NAM,
257 P 0707 ESS = NAMSC_MAXRSS,
258 P 0708 RSA = KEYINI_RES_NAM,
259 P 0709 RSS = NAMSC_MAXRSS);
260 P 0710 $RAB_INIT (RAB = KEYINI_RAB,
261 P 0711 FAB = KEYINI_FAB,
262 P 0712 RAC = SEQ);
263 P 0713
264 P 0714 ! Open the action definition file. If the open results in the RMSS_DEV error,
265 P 0715 ! it is assumed that the logical name does not exist, and success is returned.
266 P 0716 ! If the open results in the RMSS_FNF error, a warning message is issued, and
267 P 0717 ! success is returned. Any other error results in the appropriate error message
268 P 0718 ! being signaled, and the editing session terminated.
269 P 0719
270 P 0720 IF NOT $OPEN (FAB = KEYINI_FAB)
271 P 0721 THEN
272 P 0722 BEGIN
273 P 0723 IF .KEYINI_FAB[FAB$L_STS] EQL RMSS_DEV THEN RETURN 1;
274 P 0724 AED_FILERROR (AED$_INIOPENIN, KEYINI_FAB, .KEYINI_FAB[FAB$L_STS],
275 P 0725 .KEYINI_FAB[FAB$L_STV]);
276 P 0726 IF .KEYINI_FAB[FAB$L_STS] EQL RMSS_FNF THEN RETURN 1;
277 P 0727 RETURN .AED_L_WORSTERR;
278 P 0728 END;
279 P 0729 IF NOT $CONNECT (RAB = KEYINI_RAB)
280 P 0730 THEN
281 P 0731 BEGIN
282 P 0732 AED_FILERROR (AED$_INIOPENIN, KEYINI_FAB, .KEYINI_RAB[RAB$L_STS],
283 P 0733 .KEYINI_RAB[RAB$L_STV]);
284 P 0734 RETURN .AED_L_WORSTERR;
285 P 0735 END;
286 P 0736
287 P 0737 ! Loop reading the action definition file, replacing any default definition
288 P 0738 ! with those from the definition file.
289 P 0739
290 P 0740 WHILE 1
291 P 0741 DO
292 P 0742 BEGIN
293 P 0743 KEYINI_RAB[RAB$L_UBF] = DEFINE_LINE;
294 P 0744 KEYINI_RAB[RAB$L_USZ] = 512;
295 P 0745 IF NOT $GET (RAB = KEYINI_RAB)
296 P 0746 THEN
297 P 0747 BEGIN
298 P 0748 IF .KEYINI_RAB[RAB$L_STS] EQL RMSS_EOF THEN EXITLOOP;
299 P 0749 AED_FILERROR (AED$_INIREADERR, KEYINI_FAB, .KEYINI_RAB[RAB$L_STS],
300 P 0750 .KEYINI_RAB[RAB$L_STV]);
301 P 0751 RETURN .AED_L_WORSTERR;
302 P 0752 END;
303 P 0753 KEY_ACTION = 0;
304 P 0754 KEY_FLAGS = 0;
305 P 0755 KEY_STRING[DS($W_LENGTH)] = 0;
306 P 0756
307 P 0757 IF .DEFINE_LINE[0] NEQ '!'
308 P 0758 THEN
```

```
309 0759 4 BEGIN
310 0760 4 LINE_INDEX = 0;
311 0761 4 UNTIL .LINE_INDEX GEQ .KEYINI_RAB[RAB$W_RSZ]
312 0762 4 DO
313 0763 5 BEGIN
314 0764 5 IF .DEFINE_LINE[.LINE_INDEX] EQL '<'
315 0765 5 THEN
316 0766 6 BEGIN
317 0767 6 DO
318 0768 7 BEGIN
319 0769 7 LINE_INDEX = .LINE_INDEX + 1;
320 0770 7 IF .DEFINE_LINE[.LINE_INDEX] EQL '>' THEN EXITLOOP;
321 0771 7 IF .LINE_INDEX GEQ .KEYINI_RAB[RAB$W_RSZ]
322 0772 7 THEN
323 0773 8 BEGIN
324 P 0774 8 SIGNAL (AED$_DEFSYNTAX, 2, .KEYINI_RAB[RAB$W_RSZ],
325 0775 8 DEFINE_LINE);
326 0776 8 RETURN AED$_DEFSYNTAX;
327 0777 7 END;
328 0778 7 UNTIL .LINE_INDEX GEQ .KEYINI_RAB[RAB$W_RSZ];
329 0779 6 END;
330 0780 5 IF .DEFINE_LINE[.LINE_INDEX] GEQ 'a'
331 0781 5 AND .DEFINE_LINE[.LINE_INDEX] LEQ 'z'
332 0782 5 THEN DEFINE_LINE[.LINE_INDEX] = .DEFINE_LINE[.LINE_INDEX] - 32;
333 0783 5 LINE_INDEX = .LINE_INDEX + 1;
334 0784 5 END;
335 0785 4 TPARSE_BLOCK[TPASL_COUNT] = TPASK_COUNT0;
336 0786 4 TPARSE_BLOCK[TPASV_ABBREV] = 1;
337 0787 4 TPARSE_BLOCK[TPASV_BLANKS] = 1;
338 0788 4 TPARSE_BLOCK[TPASL_STRINGCNT] = .KEYINI_RAB[RAB$W_RSZ];
339 0789 4 TPARSE_BLOCK[TPASL_STRINGPTR] = DEFINE_LINE;
340 0790 4
341 0791 4 LOCAL STATUS = LIB$TPARSE (TPARSE_BLOCK, KEYDEF_STATE, KEYDEF_KEY);
342 0792 4 IF NOT .LOCAL_STATUS
343 0793 4 THEN
344 0794 5 BEGIN
345 0795 5 SIGNAL (AED$_DEFSYNTAX, 2, .TPARSE_BLOCK[TPASL_STRINGCNT],
346 P 0796 5 .TPARSE_BLOCK[TPASL_STRINGPTR]);
347 0797 5 RETURN AED$_DEFSYNTAX;
348 0798 5 END;
349 0799 4 END;
350 0800 3 END;
351 0801 2 END;
352 0802 2 RETURN 1;
353 0803 2
354 0804 2
355 0805 1 END;
```

! End of routine AED_GETKEYINI

.TITLE AED\$DECODE
.IDENT \V04-000\

.PSECT _LIB\$KEY1\$,NOWRT, SHR, PIC,1

00000 ;TPASKEYSTO
0.4: ;BLKB 0
45 4E 49 46 45 44 00000 ;TPASKEYST


```
001F6 :TPASKEYSTO
33 53 53 U.277: .BLKB 0
001F6 :TPASKEYST
FF U.279: .ASCII \SS3\
001F9 :TPASKEYSTO
001FA :TPASKEYSTO
54 55 4F 42 55 52 U.284: .BLKB 0
001FA :TPASKEYST
FF U.286: .ASCII \RUBOUT\
00200 :TPASKEYSTO
FF U.288: .ASCII \RUBOUT\
00201 :TPASKEYFILL
00202 :TPASKEYSTO
52 4F 00202 U.291: .BYTE -1
00202 :TPASKEYSTO
FF U.296: .BLKB 0
00202 :TPASKEYST
FF U.298: .ASCII \OR\
00204 :TPASKEYSTO
FF U.300: .ASCII \OR\
00205 :TPASKEYFILL
U.305: .BYTE -1

.PSECT _LIB$STATES, NOWRT, SHR, PIC, 1

00000 KEYDEF_STATE::
00000 SWALLOW_1: .BLKB 0
11F2 00000 :TPASTYPE
U.2: .WORD 4594
0000* 00002 :TPASTARGET
U.3: .WORD <<SWALLOW_1-U.3>-2>
0500 00004 :TPASTYPE
U.7: .WORD 1280
00006 SWALLOW_2: .BLKB 0
11F2 00006 :TPASTYPE
U.9: .WORD 4594
0000* 00008 :TPASTARGET
U.10: .WORD <<SWALLOW_2-U.10>-2>
6101 0000A :TPASTYPE
U.14: .WORD 24833
00000000* 0000C :TPASADDR
U.15: .LONG <<KEY_ACTION-U.15>-4>
00000001 00010 :TPASMASK
U.16: .LONG 1
6102 00014 :TPASTYPE
U.20: .WORD 24834
00000000* 00016 :TPASADDR
U.21: .LONG <<KEY_ACTION-U.21>-4>
00000002 0001A :TPASMASK
U.22: .LONG 2
6103 0001E :TPASTYPE
U.26: .WORD 24835
00000000* 00020 :TPASADDR
U.27: .LONG <<KEY_ACTION-U.27>-4>
00000003 00024 :TPASMASK
U.28: .LONG 3
6104 00028 :TPASTYPE
```

00000000*	0002A	U.32: .WORD	24836	:
		:TPASADDR		:
00000004	0002E	U.33: .LONG	<<KEY_ACTION-U.33>-4>	:
		:TPASMASK		:
6105	00032	U.34: .LONG	4	:
		:TPASTYPE		:
		U.38: .WORD	24837	:
00000000*	00034	:TPASADDR		:
		U.39: .LONG	<<KEY_ACTION-U.39>-4>	:
00000005	00038	:TPASMASK		:
		U.40: .LONG	5	:
6106	0003C	:TPASTYPE		:
		U.44: .WORD	24838	:
00000000*	0003E	:TPASADDR		:
		U.45: .LONG	<<KEY_ACTION-U.45>-4>	:
00000006	00042	:TPASMASK		:
		U.46: .LONG	6	:
6107	00046	:TPASTYPE		:
		U.50: .WORD	24839	:
00000000*	00048	:TPASADDR		:
		U.51: .LONG	<<KEY_ACTION-U.51>-4>	:
00000007	0004C	:TPASMASK		:
		U.52: .LONG	7	:
6108	00050	:TPASTYPE		:
		U.56: .WORD	24840	:
00000000*	00052	:TPASADDR		:
		U.57: .LONG	<<KEY_ACTION-U.57>-4>	:
00000008	00056	:TPASMASK		:
		U.58: .LONG	8	:
6109	0005A	:TPASTYPE		:
		U.62: .WORD	24841	:
00000000*	0005C	:TPASADDR		:
		U.63: .LONG	<<KEY_ACTION-U.63>-4>	:
00000009	00060	:TPASMASK		:
		U.64: .LONG	9	:
610A	00064	:TPASTYPE		:
		U.68: .WORD	24842	:
00000000*	00066	:TPASADDR		:
		U.69: .LONG	<<KEY_ACTION-U.69>-4>	:
0000000A	0006A	:TPASMASK		:
		U.70: .LONG	10	:
610B	0006E	:TPASTYPE		:
		U.74: .WORD	24843	:
00000000*	00070	:TPASADDR		:
		U.75: .LONG	<<KEY_ACTION-U.75>-4>	:
0000000B	00074	:TPASMASK		:
		U.76: .LONG	11	:
610C	00078	:TPASTYPE		:
		U.80: .WORD	24844	:
00000000*	0007A	:TPASADDR		:
		U.81: .LONG	<<KEY_ACTION-U.81>-4>	:
0000000C	0007E	:TPASMASK		:
		U.82: .LONG	12	:
610D	00082	:TPASTYPE		:
		U.86: .WORD	24845	:
00000000*	00084	:TPASADDR		:
		U.87: .LONG	<<KEY_ACTION-U.87>-4>	:

0000000E	00088	:TPASMASK	
		U.88:	LONG 14
610E	0008C	:TPASTYPE	
		U.92:	WORD 24846
00000000*	0008E	:TPASADDR	
		U.93:	LONG <<KEY_ACTION-U.93>-4>
00000010	00092	:TPASMASK	
		U.94:	LONG 16
610F	00096	:TPASTYPE	
		U.98:	WORD 24847
00000000*	00098	:TPASADDR	
		U.99:	LONG <<KEY_ACTION-U.99>-4>
00000011	0009C	:TPASMASK	
		U.100:	LONG 17
6110	000A0	:TPASTYPE	
		U.104:	WORD 24848
00000000*	000A2	:TPASADDR	
		U.105:	LONG <<KEY_ACTION-U.105>-4>
00000012	000A6	:TPASMASK	
		U.106:	LONG 18
6111	000AA	:TPASTYPE	
		U.110:	WORD 24849
00000000*	000AC	:TPASADDR	
		U.111:	LONG <<KEY_ACTION-U.111>-4>
00000013	000B0	:TPASMASK	
		U.112:	LONG 19
6112	000B4	:TPASTYPE	
		U.116:	WORD 24850
00000000*	000B6	:TPASADDR	
		U.117:	LONG <<KEY_ACTION-U.117>-4>
00000014	000BA	:TPASMASK	
		U.118:	LONG 20
6113	000BE	:TPASTYPE	
		U.122:	WORD 24851
00000000*	000C0	:TPASADDR	
		U.123:	LONG <<KEY_ACTION-U.123>-4>
00000015	000C4	:TPASMASK	
		U.124:	LONG 21
6114	000C8	:TPASTYPE	
		U.128:	WORD 24852
00000000*	000CA	:TPASADDR	
		U.129:	LONG <<KEY_ACTION-U.129>-4>
00000016	000CE	:TPASMASK	
		U.130:	LONG 22
6115	000D2	:TPASTYPE	
		U.134:	WORD 24853
00000000*	000D4	:TPASADDR	
		U.135:	LONG <<KEY_ACTION-U.135>-4>
00000017	000D8	:TPASMASK	
		U.136:	LONG 23
6116	000DC	:TPASTYPE	
		U.140:	WORD 24854
00000000*	000DE	:TPASADDR	
		U.141:	LONG <<KEY_ACTION-U.141>-4>
00000018	000E2	:TPASMASK	
		U.142:	LONG 24
6117	000E6	:TPASTYPE	

00000000*	000E8	U.146: .WORD	24855	:
		:TPASADDR		:
00000019	000EC	U.147: .LONG	<<KEY_ACTION-U.147>-4>	:
		:TPASMASK		:
6118	000F0	U.148: .LONG	25	:
		:TPASTYPE		:
00000000*	000F2	U.152: .WORD	24856	:
		:TPASADDR		:
0000001A	000F6	U.153: .LONG	<<KEY_ACTION-U.153>-4>	:
		:TPASMASK		:
6119	000FA	U.154: .LONG	26	:
		:TPASTYPE		:
00000000*	000FC	U.158: .WORD	24857	:
		:TPASADDR		:
0000001B	00100	U.159: .LONG	<<KEY_ACTION-U.159>-4>	:
		:TPASMASK		:
611A	00104	U.160: .LONG	27	:
		:TPASTYPE		:
00000000*	00106	U.164: .WORD	24858	:
		:TPASADDR		:
0000001C	0010A	U.165: .LONG	<<KEY_ACTION-U.165>-4>	:
		:TPASMASK		:
611B	0010E	U.166: .LONG	28	:
		:TPASTYPE		:
00000000*	00110	U.170: .WORD	24859	:
		:TPASADDR		:
0000001D	00114	U.171: .LONG	<<KEY_ACTION-U.171>-4>	:
		:TPASMASK		:
611C	00118	U.172: .LONG	29	:
		:TPASTYPE		:
00000000*	0011A	U.176: .WORD	24860	:
		:TPASADDR		:
0000001E	0011E	U.177: .LONG	<<KEY_ACTION-U.177>-4>	:
		:TPASMASK		:
611D	00122	U.178: .LONG	30	:
		:TPASTYPE		:
00000000*	00124	U.182: .WORD	24861	:
		:TPASADDR		:
0000001F	00128	U.183: .LONG	<<KEY_ACTION-U.183>-4>	:
		:TPASMASK		:
611E	0012C	U.184: .LONG	31	:
		:TPASTYPE		:
00000000*	0012E	U.188: .WORD	24862	:
		:TPASADDR		:
00000021	00132	U.189: .LONG	<<KEY_ACTION-U.189>-4>	:
		:TPASMASK		:
611F	00136	U.190: .LONG	33	:
		:TPASTYPE		:
00000000*	00138	U.194: .WORD	24863	:
		:TPASADDR		:
00000022	0013C	U.195: .LONG	<<KEY_ACTION-U.195>-4>	:
		:TPASMASK		:
6120	00140	U.196: .LONG	34	:
		:TPASTYPE		:
00000000*	00142	U.200: .WORD	24864	:
		:TPASADDR		:
		U.201: .LONG	<<KEY_ACTION-U.201>-4>	:

00000025	00146	:TPASMASK	
		U.202:	LONG 37
6121	0014A	:TPASTYPE	
		U.206:	WORD 24865
00000000*	0014C	:TPASADDR	
		U.207:	LONG <<KEY_ACTION-U.207>-4>
00000026	00150	:TPASMASK	
		U.208:	LONG 38
6122	00154	:TPASTYPE	
		U.212:	WORD 24866
00000000*	00156	:TPASADDR	
		U.213:	LONG <<KEY_ACTION-U.213>-4>
00000023	0015A	:TPASMASK	
		U.214:	LONG 35
6123	0015E	:TPASTYPE	
		U.218:	WORD 24867
00000000*	00160	:TPASADDR	
		U.219:	LONG <<KEY_ACTION-U.219>-4>
00000024	00164	:TPASMASK	
		U.220:	LONG 36
6124	00168	:TPASTYPE	
		U.224:	WORD 24868
00000000*	0016A	:TPASADDR	
		U.225:	LONG <<KEY_ACTION-U.225>-4>
00000027	0016E	:TPASMASK	
		U.226:	LONG 39
6125	00172	:TPASTYPE	
		U.230:	WORD 24869
00000000*	00174	:TPASADDR	
		U.231:	LONG <<KEY_ACTION-U.231>-4>
00000028	00178	:TPASMASK	
		U.232:	LONG 40
6526	0017C	:TPASTYPE	
		U.236:	WORD 25894
00000000*	0017E	:TPASADDR	
		U.237:	LONG <<KEY_ACTION-U.237>-4>
00000029	00182	:TPASMASK	
		U.238:	LONG 41
	00186	SWALLOW_3:	
		BLKB	0
11F2	00186	:TPASTYPE	
		U.240:	WORD 4594
0000*	00188	:TPATARGET	
		U.241:	WORD <<SWALLOW_3-U.241>-2>
0527	0018A	:TPASTYPE	
		U.245:	WORD 1319
	0018C	KEY_DEFINE:	
		BLKB	0
11F2	0018C	:TPASTYPE	
		U.247:	WORD 4594
0000*	0018E	:TPATARGET	
		U.248:	WORD <<KEY_DEFINE-U.248>-2>
6128	00190	:TPASTYPE	
		U.252:	WORD 24872
00000000*	00192	:TPASADDR	
		U.253:	LONG <<KEY_FLAGS-U.253>-4>
00000004	00196	:TPASMASK	

7129	0019A	U.254: .LONG	4	:
		:TPASTYPE		:
00000000*	0019C	U.258: .WORD	28969	:
		:TPASADDR		:
00000008	001A0	U.259: .LONG	<<KEY_FLAGS-U.259>-4>	:
		:TPASMASK		:
0000*	001A4	U.260: .LONG	8	:
		:TPASTARGET		:
712A	001A6	U.262: .WORD	<<U.261-U.262>-2>	:
		:TPASTYPE		:
00000000*	001A8	U.266: .WORD	28970	:
		:TPASADDR		:
00000010	001AC	U.267: .LONG	<<KEY_FLAGS-U.267>-4>	:
		:TPASMASK		:
0000*	001B0	U.268: .LONG	16	:
		:TPASTARGET		:
712B	001B2	U.269: .WORD	<<U.261-U.269>-2>	:
		:TPASTYPE		:
00000000*	001B4	U.273: .WORD	28971	:
		:TPASADDR		:
00000001	001B8	U.274: .LONG	<<KEY_FLAGS-U.274>-4>	:
		:TPASMASK		:
0000*	001BC	U.275: .LONG	1	:
		:TPASTARGET		:
712C	001BE	U.276: .WORD	<<U.261-U.276>-2>	:
		:TPASTYPE		:
00000000*	001C0	U.280: .WORD	28972	:
		:TPASADDR		:
00000002	001C4	U.281: .LONG	<<KEY_FLAGS-U.281>-4>	:
		:TPASMASK		:
0000*	001C8	U.282: .LONG	2	:
		:TPASTARGET		:
812D	001CA	U.283: .WORD	<<U.261-U.283>-2>	:
		:TPASTYPE		:
00000000V	001CC	U.287: .WORD	-32467	:
		:TPASACTION		:
15F7	001D0	U.288: .LONG	<<SET_RUBOUT-U.288>-4>	:
		:TPASTYPE		:
FFFE	001D2	U.289: .WORD	5623	:
		:TPASTARGET		:
	001D4	U.290: .WORD	-2	:
		CHECK_END:		:
11F2	001D4	.BLKB	0	:
		:TPASTYPE		:
0000*	001D6	U.292: .WORD	4594	:
		:TPASTARGET		:
102C	001D8	U.293: .WORD	<<CHECK_END-U.293>-2>	:
		:TPASTYPE		:
0000*	001DA	U.294: .WORD	4140	:
		:TPASTARGET		:
912E	001DC	U.295: .WORD	<<KEY_DEFINE-U.295>-2>	:
		:TPASTYPE		:
00000000V	001DE	U.299: .WORD	-28370	:
		:TPASACTION		:
0000*	001E2	U.300: .LONG	<<SET_DEFINITION-U.300>-4>	:
		:TPASTARGET		:
		U.301: .WORD	<<KEY_DEFINE-U.301>-2>	:

95F7	001E4	:TPATYPE			
		U.302:	WORD	-27145	:
00000000V	001E6	:TPASACTION			:
		U.303:	LONG	<<SET_DEFINITION-U.303>-4>	:
FFFF	001EA	:TPATARGET			:
		U.304:	WORD	-1	:
	001EC	:GET TEXT			:
		U.26T:	BLKB	0	:
11F2	001EC	:TPATYPE			:
		U.306:	WORD	4594	:
0000*	001EE	:TPATARGET			:
		U.307:	WORD	<<U.261-U.307>-2>	:
042C	001F0	:TPATYPE			:
		U.308:	WORD	1068	:
	001F2	SWALLOW_4:			:
		BLKB	0		:
11F2	001F2	:TPATYPE			:
		U.309:	WORD	4594	:
0000*	001F4	:TPATARGET			:
		U.310:	WORD	<<SWALLOW_4-U.310>-2>	:
5DF8	001F6	:TPATYPE			:
		U.311:	WORD	24056	:
0000*	001F8	:TPASUBEXP			:
		U.313:	WORD	<<U.312-U.313>-2>	:
00000000*	001FA	:TPASADDR			:
		U.314:	LONG	<<KEY_STRING-U.314>-4>	:
0000*	001FE	:TPATARGET			:
		U.315:	WORD	<<CHECK_END-U.315>-2>	:
	00200	:GET STRING			:
		U.312:	BLKB	0	:
19F8	00200	:TPATYPE			:
		U.316:	WORD	6648	:
0000*	00202	:TPASUBEXP			:
		U.318:	WORD	<<U.317-U.318>-2>	:
0000*	00204	:TPATARGET			:
		U.319:	WORD	<<U.312-U.319>-2>	:
15F6	00206	:TPATYPE			:
		U.320:	WORD	5622	:
FFFF	00208	:TPATARGET			:
		U.321:	WORD	-1	:
	0020A	:CHECK_DELIM			:
		U.317:	BLKB	0	:
102C	0020A	:TPATYPE			:
		U.322:	WORD	4140	:
FFFE	0020C	:TPATARGET			:
		U.323:	WORD	-2	:
1020	0020E	:TPATYPE			:
		U.324:	WORD	4128	:
FFFE	00210	:TPATARGET			:
		U.325:	WORD	-2	:
11F7	00212	:TPATYPE			:
		U.326:	WORD	4599	:
FFFE	00214	:TPATARGET			:
		U.327:	WORD	-2	:
15ED	00216	:TPATYPE			:
		U.328:	WORD	5613	:
FFFF	00218	:TPATARGET			:

```
U.329: .WORD -1
.PSECT _LIB$KEY0$,NOWRT, SHR, PIC,1
00000 KEYDEF_KEY::
00000 :TPASKEY0 .BLKB 0
0000* 00000 U.1: .BLKB 0
0000* 00000 :TPASKEY
0000* 00002 U.5: .WORD <U.4-U.1>
0000* 00002 :TPASKEY
0000* 00004 U.12: .WORD <U.11-U.1>
0000* 00004 :TPASKEY
0000* 00006 U.18: .WORD <U.17-U.1>
0000* 00006 :TPASKEY
0000* 00008 U.24: .WORD <U.23-U.1>
0000* 00008 :TPASKEY
0000* 0000A U.30: .WORD <U.29-U.1>
0000* 0000A :TPASKEY
0000* 0000C U.36: .WORD <U.35-U.1>
0000* 0000C :TPASKEY
0000* 0000E U.42: .WORD <U.41-U.1>
0000* 0000E :TPASKEY
0000* 00010 U.48: .WORD <U.47-U.1>
0000* 00010 :TPASKEY
0000* 00012 U.54: .WORD <U.53-U.1>
0000* 00012 :TPASKEY
0000* 00014 U.60: .WORD <U.59-U.1>
0000* 00014 :TPASKEY
0000* 00016 U.66: .WORD <U.65-U.1>
0000* 00016 :TPASKEY
0000* 00018 U.72: .WORD <U.71-U.1>
0000* 00018 :TPASKEY
0000* 0001A U.78: .WORD <U.77-U.1>
0000* 0001A :TPASKEY
0000* 0001C U.84: .WORD <U.83-U.1>
0000* 0001C :TPASKEY
0000* 0001E U.90: .WORD <U.89-U.1>
0000* 0001E :TPASKEY
0000* 00020 U.96: .WORD <U.95-U.1>
0000* 00020 :TPASKEY
0000* 00022 U.102: .WORD <U.101-U.1>
0000* 00022 :TPASKEY
0000* 00024 U.108: .WORD <U.107-U.1>
0000* 00024 :TPASKEY
0000* 00026 U.114: .WORD <U.113-U.1>
0000* 00026 :TPASKEY
0000* 00028 U.120: .WORD <U.119-U.1>
0000* 00028 :TPASKEY
0000* 0002A U.126: .WORD <U.125-U.1>
0000* 0002A :TPASKEY
0000* 0002C U.132: .WORD <U.131-U.1>
0000* 0002C :TPASKEY
0000* 0002E U.138: .WORD <U.137-U.1>
0000* 0002E :TPASKEY
0000* 00030 U.144: .WORD <U.143-U.1>
0000* 00030 :TPASKEY
```

1
15-Sep-1984 23:37:58
14-Sep-1984 11:52:23VAX-11 Bliss-32 V4.0-742
[ACLEDT.SRC]AEDDECODE.B32;1Page 21
(3)

0000*	00032	U.150: .WORD	<U.149-U.1>	:
		:TPASKEY		:
0000*	00034	U.156: .WORD	<U.155-U.1>	:
		:TPASKEY		:
0000*	00036	U.162: .WORD	<U.161-U.1>	:
		:TPASKEY		:
0000*	00038	U.168: .WORD	<U.167-U.1>	:
		:TPASKEY		:
0000*	0003A	U.174: .WORD	<U.173-U.1>	:
		:TPASKEY		:
0000*	0003C	U.180: .WORD	<U.179-U.1>	:
		:TPASKEY		:
0000*	0003E	U.186: .WORD	<U.185-U.1>	:
		:TPASKEY		:
0000*	00040	U.192: .WORD	<U.191-U.1>	:
		:TPASKEY		:
0000*	00042	U.198: .WORD	<U.197-U.1>	:
		:TPASKEY		:
0000*	00044	U.204: .WORD	<U.203-U.1>	:
		:TPASKEY		:
0000*	00046	U.210: .WORD	<U.209-U.1>	:
		:TPASKEY		:
0000*	00048	U.216: .WORD	<U.215-U.1>	:
		:TPASKEY		:
0000*	0004A	U.222: .WORD	<U.221-U.1>	:
		:TPASKEY		:
0000*	0004C	U.228: .WORD	<U.227-U.1>	:
		:TPASKEY		:
0000*	0004E	U.234: .WORD	<U.233-U.1>	:
		:TPASKEY		:
0000*	00050	U.243: .WORD	<U.242-U.1>	:
		:TPASKEY		:
0000*	00052	U.250: .WORD	<U.249-U.1>	:
		:TPASKEY		:
0000*	00054	U.256: .WORD	<U.255-U.1>	:
		:TPASKEY		:
0000*	00056	U.264: .WORD	<U.263-U.1>	:
		:TPASKEY		:
0000*	00058	U.271: .WORD	<U.270-U.1>	:
		:TPASKEY		:
0000*	0005A	U.278: .WORD	<U.277-U.1>	:
		:TPASKEY		:
0000*	0005C	U.285: .WORD	<U.284-U.1>	:
		:TPASKEY		:
		U.297: .WORD	<U.296-U.1>	:

.PSECT AED_COMMON,NOEXE, OVR,0

00000	AED_L_FLAGS:	
	.BLKB	4
00004	AED_B_OPTIONS:	
	.BLKB	1
00005		
	.BLKB	3
00008	AED_L_OBJTYP:	
	.BLKB	4
0000C	AED_Q_OBJNAM:	
	.BLKB	8


```

00014 AED_L_WORSTERR:
      .BLKB 4
00018 AED_L_PAGEWIDTH:
      .BLKB 4
0001C AED_L_PAGESIZE:
      .BLKB 4
00020 AED_B_COLUMN:
      .BLKB 1
00021      .BLKB 3
00024 AED_B_LINE:
      .BLKB 1
00025      .BLKB 3
00028 AED_B_SAVE_COL:
      .BLKB 1
00029      .BLKB 3
0002C AED_B_SAVE_LIN:
      .BLKB 1
0002D      .BLKB 3
00030 AED_Q_LINETABLE:
      .BLKB 12
0003C AED_L_CURACE:
      .BLKB 4
00040 AED_L_FIRSTLINE:
      .BLKB 4
00044 AED_L_LASTLINE:
      .BLKB 4
00048 AED_L_BEGINLINE:
      .BLKB 4
0004C AED_W_INPUTLEN:
      .BLKB 2
0004E      .BLKB 2
00050 AED_Q_DEL ACE:
      .BLKB 8
00058 AED_Q_DEL LINE:
      .BLKB 8
00060 AED_Q_DEL WORD:
      .BLKB 8
00068 AED_B_DEL CHAR:
      .BLKB 1
00069      .BLKB 3
0006C AED_A_ACLBUFFER:
      .BLKB 4
00070 AED_Q_OUTLINE:
      .BLKB 8
00078 AED_W_OBJCHAN:
      .BLKB 2
0007A      .BLKB 2
0007C AED_W_TERMIN:
      .BLKB 2
0007E      .BLKB 2
00080 AED_W_TERMOUT:
      .BLKB 2
00082      .BLKB 2
00084 AED_W_IOSB:
      .BLKB 8
0008C AED_L_STATUS:
      .BLKB 4

```

L 1
13-Sep-1984 23:37:58
14-Sep-1984 11:52:23VAX-11 BLISS-32 V4.0-742
[ACLEDIT.SRC]AEDDECODE.B32:1Page 23
(3)

```
00090 AED_B_FIELD:
      .BLKB 1
00091      .BLKB 3
00094 AED_W_FIELDBEG:
      .BLKB 2
00096      .BLKB 2
00098 AED_W_FIELDEND:
      .BLKB 2
0009A      .BLKB 2
0009C AED_B_ITEM:
      .BLKB 1
0009D      .BLKB 3
000A0 AED_W_ITEMBEG:
      .BLKB 2
000A2      .BLKB 2
000A4 AED_W_ITEMEND:
      .BLKB 2
000A6      .BLKB 2
000A8 AED_B_ACETYPE:
      .BLKB 1
000A9      .BLKB 3
000AC AED_W_JOURNAL:
      .BLKB 2
000AE      .BLKB 2
000B0 AED_T_CURLINE:
      .BLKB 532
002C4 AED_W_TOTALSIZE:
      .BLKB 2
002C6      .BLKB 2
002C8 JOURNAL_FAB:
      .BLKB 80
00318 JOURNAL_NAM:
      .BLKB 96
00378 JOURNAL_RAB:
      .BLKB 68
003BC JOURNAL_XABPRO:
      .BLKB 88
00414 JOURNAL_BUFFER:
      .BLKB 10
0041E      .BLKB 2
00420 JOURNAL_INDEX:
      .BLKB 4
00424 RECOVER_FAB:
      .BLKB 80
00474 RECOVER_NAM:
      .BLKB 96
004D4 RECOVER_RAB:
      .BLKB 68
00518 RECOVER_BUFFER:
      .BLKB 10
00522      .BLKB 2
00524 RECOVER_INDEX:
      .BLKB 4
```

.PSECT \$SPLITS,NOWRT,NOEXE,2

00 00 3A 54 49 4E 49 24 54 49 44 45 4C 43 41 00000 P.AAA: .ASCII \ACLEDIT\$INIT:\<0><0><0>

;

00 0000F

.PSECT \$OWNS,NOEXE,2

00000 KEY_BLOCK:

.BLKB 11

0000B .BLKB 1

0000C KEY_STRING:

.BLKB 8

KEY_ACTION=

KEY_BLOCK+8

KEY_FLAGS=

KEY_BLOCK+10

.EXTRN CLISGET-VALUE, CLISPRESENT
.EXTRN LIB\$FREE_VM, LIB\$GET_VM
.EXTRN LIB\$PARSE, SCR\$DOWN_SCROLL
.EXTRN SCR\$ERASE_LINE, SCR\$ERASE_PAGE
.EXTRN SCR\$SET_CURSOR, SCR\$SET_SCROLL
.EXTRN SCR\$UP_SCROLL, AED\$OBJLOCKED
.EXTRN AED\$BADKEEP, AED\$LOCATERR
.EXTRN AED\$INIREADERR
.EXTRN AED\$JOUWRITERR
.EXTRN AED\$JOUOPENOUT
.EXTRN AED\$JOUCLOSEOUT
.EXTRN AED\$RECREADERR
.EXTRN AED\$REOPENIN, AED\$RECLOSEIN
.EXTRN AED\$BADJIC, AED\$BADGRPMEM
.EXTRN AED\$SYNTAX, AED\$BADTYPE
.EXTRN AED\$NOITEMSEL, AED\$MUSTENTER
.EXTRN AED\$INIOPENIN, AED\$INICLOSIN
.EXTRN AED\$DEFSYNTAX, AED\$NODELETE
.EXTRN AED\$NOMODIFY, AED\$NOHIDDEN
.EXTRN AED\$DUPLICATE, AED\$NOCOMBINE
.EXTRN AED\$NODEFAULT, AED\$NOCTRLCHAR
.EXTRN AED\$NOTFOUND, AED\$CONTROL_C
.EXTRN AED\$ACLUPDATED
.EXTRN AED\$NOCHANGE, AED\$FILEERROR
.EXTRN AED\$PUTOUTPUT, AED\$SET_CURSOR
.EXTRN KEY_TABLE, SYS\$OPEN
.EXTRN SYS\$CONNECT, SYS\$GET
.EXTRN LIB\$SIGNAL

.PSECT \$CODE\$,NOWRT,2

OFFC 00000

.ENTRY AED_GETKEYINI, Save R2,R3,R4,R5,R6,R7,R8,-
R9,R10,R11
MOVAB LIB\$SIGNAL, R11
MOVAB SCR\$ERASE_PAGE, R10
MOVL #AED\$INIOPENIN, R9
MOVL #AED\$DEFSYNTAX, R8
MOVAB SCR\$SET_CURSOR, R7
MOVAB AED\$L_WORSTERR, R6
MOVAB -1304(SP), SP
MOVCS #0, (SP), #0, #80, \$RMS_PTR

MOVW #20483, \$RMS_PTR
MOVZBL #64, \$RMS_PTR+4
MOVB #2, \$RMS_PTR+22

0645

0704

0050 8F

00

5B	00000000G	00	9E	00002
5A	00000000G	00	9E	00009
59	00000000G	8F	D0	00010
58	00000000G	8F	D0	00017
57	00000000G	00	9E	0001E
56	0000	CF	9E	00025
5E	FAE8	CE	9E	0002A
6E		00	2C	0002F
	B0	AD		00036
B0	AD	5003	8F	B0 00038
B4	AD	40	8F	9A 0003E
C6	AD		02	90 00043

Address	Hex	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

52	8E	AD	10	0083	31	00143	BRW	17\$			
			3C	24	AE42	91	00146	8\$:	CMPB	DEFINE_LINE[LINE_INDEX], #60	
					62	12	0014B		BNEQ	15\$	
			3E	24	AE42	06	0014D	9\$:	INCL	LINE_INDEX	
					52	91	0014F		CMPB	DEFINE_LINE[LINE_INDEX], #62	
					59	13	00154		BEQL	15\$	
					53	D4	00156		CLRL	R3	
			10		00	ED	00158		CMPZV	#0, #16, KEYINI_RAB+34, LINE_INDEX	
					4C	14	0015E		BGTR	14\$	
					53	D6	00160		INCL	R3	
		OE	EC	A6	03	E1	00162		BBC	#3, AED_L_FLAGS, 10\$	
					01	DD	00167		PUSHL	#1	
					15	DD	00169		PUSHL	#21	
			6A		02	FB	0016B		CALLS	#2, SCR\$ERASE_PAGE	
					01	DD	0016E		PUSHL	#1	
					15	DD	00170		PUSHL	#21	
			67		02	FB	00172		CALLS	#2, SCR\$SET_CURSOR	
			7E	24	AE	9F	00175	10\$:	PUSHAB	DEFINE_LINE	
				8E	AD	3C	00178		MOVZWL	KEYINI_RAB+34, -(SP)	
					02	DD	0017C		PUSHL	#2	
					58	DD	0017E		PUSHL	R8	
			6B		04	FB	00180		CALLS	#4, LIB\$SIGNAL	
		OB	EC	A6	03	E1	00183		BBC	#3, AED_L_FLAGS, 11\$	
				7E	OC	A6	00188		MOVZBL	AED_B_COLUMN, -(SP)	
				7E	10	A6	0018C		MOVZBL	AED_B_LINE, -(SP)	
				67		02	FB	00190	CALLS	#2, SCR\$SET_CURSOR	
				00000000*	8F	D5	00193	11\$:	TSTL	#<AED\$_DEFSYNTAX&7>	
					03	12	00199		BNEQ	13\$	
					009D	31	0019B	12\$:	BRW	22\$	
00000000*	8F		66		00	ED	0019E	13\$:	CMPZV	#0, #3, AED_L_WORSTERR, #<AED\$_DEFSYNTAX&7>	
					F2	18	001A7		BGEQ	12\$	
					008C	31	001A9		BRW	21\$	
			9E		53	E9	001AC	14\$:	BLBC	R3, 9\$	
			61	8F	24	AE42	91	001AF	15\$:	CMPB	DEFINE_LINE[LINE_INDEX], #97
					0D	1F	001B5		BLSSU	16\$	
			7A	8F	24	AE42	91	001B7		CMPB	DEFINE_LINE[LINE_INDEX], #122
					05	1A	001BD		BGTRU	16\$	
			24	AE42		20	82	001BF		SUBB2	#32, DEFINE_LINE[LINE_INDEX]
					52	D6	001C4	16\$:	INCL	LINE_INDEX	
					FF72	31	001C6		BRW	7\$	
					08	D0	001C9	17\$:	MOVL	#8, TPARSE_BLOCK	
			6E		03	88	001CC		BISB2	#3, TPARSE_BLOCK+4	
		04	AE		8E	AD	3C	001D0		MOVZWL	KEYINI_RAB+34, TPARSE_BLOCK+8
		08	AE		24	AE	9E	001D5		MOVAB	DEFINE_LINE, TPARSE_BLOCK+12
		OC	AE		0000*	CF	9F	001DA		PUSHAB	KEYDEF_KEY
					0000*	CF	9F	001DE		PUSHAB	KEYDEF_STATE
					08	AE	9F	001E2		PUSHAB	TPARSE_BLOCK
			00000000G	00	03	FB	001E5		CALLS	#3, LIB\$TPARSE	
				54	50	D0	001EC		MOVL	R0, LOCAL_STATUS	
				03	54	E9	001EF		BLBC	LOCAL_STATUS, 18\$	
					FEF7	31	001F2		BRW	3\$	
		OE	EC	A6	03	E1	001F5	18\$:	BBC	#3, AED_L_FLAGS, 19\$	
					01	DD	001FA		PUSHL	#1	
					15	DD	001FC		PUSHL	#21	
			6A		02	FB	001FE		CALLS	#2, SCR\$ERASE_PAGE	
					01	DD	00201		PUSHL	#1	
					15	DD	00203		PUSHL	#21	

AED\$DECODE
V04-000

C 2
15-Sep-1984 23:37:58
14-Sep-1984 11:52:23

VAX-11 BLISS-32 V4.0-742
[ACLEDT.SRC]AEDDECODE.B32;1

Page 27
(3)

		67		02	FB	00205		CALLS	#2, SCR\$SET_CURSOR	
			OC	AE	DD	00208	19\$:	PUSHL	TPARSE_BLOCK+12	
			OC	AE	DD	0020B		PUSHL	TPARSE_BLOCK+8	
				02	DD	0020E		PUSHL	#2	
				58	DD	00210		PUSHL	R8	
		6B		04	FB	00212		CALLS	#4, LIB\$SIGNAL	
	0B	A6	EC	03	E1	00215		BBC	#3, AED_L_FLAGS, 20\$	
		7E		A6	9A	0021A		MOVZBL	AED_B_COLUMN, -(SP)	
		7E		A6	9A	0021E		MOVZBL	AED_B_LINE, -(SP)	
		67		02	FB	00222		CALLS	#2, SCR\$SET_CURSOR	
				8F	D5	00225	20\$:	TSTL	#<AED\$_DEFSYNTAX&7>	
			00000000*	0E	13	0022B		BEQL	22\$	
00000000*	8F			00	ED	0022D		CMPZV	#0, #3, AED_L_WORSTERR, #<AED\$_DEFSYNTAX&7>	
		66		03	18	00236		BGEQ	22\$	
		50		58	D0	00238	21\$:	MOVL	R8, AED_L_WORSTERR	
				58	D0	0023B	22\$:	MOVL	R8, R0	0798
					04	0023E		RET		
		50		01	D0	0023F	23\$:	MOVL	#1, R0	0803
					04	00242		RET		0805

; Routine Size: 579 bytes, Routine Base: \$CODE\$ + 0000


```
0806 1 ROUTINE SET RUBOUT =
0807 1
0808 1 ++
0809 1
0810 1 FUNCTIONAL DESCRIPTION:
0811 1
0812 1 This routine sets up the string descriptor to point to a single
0813 1 rubout character.
0814 1
0815 1 CALLING SEQUENCE:
0816 1 SET RUBOUT ( )
0817 1
0818 1 INPUT PARAMETERS:
0819 1 none
0820 1
0821 1 IMPLICIT INPUTS:
0822 1 none
0823 1
0824 1 OUTPUT PARAMETERS:
0825 1 none
0826 1
0827 1 IMPLICIT OUTPUTS:
0828 1 KEY_STRING: descriptor to action defining string
0829 1
0830 1 ROUTINE VALUE:
0831 1 1
0832 1
0833 1 SIDE EFFECTS:
0834 1 none
0835 1
0836 1 --
0837 1
0838 2 BEGIN
0839 2
0840 2 KEY_STRING[DSC$W_LENGTH] = 1;
0841 2 KEY_STRING[DSC$A_POINTER] = UPLIT BYTE (%CHAR (%X'7F'));
0842 2
0843 2 RETURN 1;
0844 2
0845 1 END;

! End of routine SET RUBOUT
```

.PSECT \$PLITS,NOWRT,NOEXE,2

7F 00010 P.AAB: .ASCII <127>

.PSECT \$CODE\$,NOWRT,2

0000 00000 SET RUBOUT:

0000*	CF	01	80	00002	.WORD	Save nothing	0806
0000*	CF	01	9E	00007	MOVW	#1, KEY_STRING	0840
	SD	01	D0	0000E	MOVAB	P.AAB, KEY_STRING+4	0841
			04	00011	MOVL	#1, R0	0843
					RET		0845

AED\$DECODE
V04-000

E 2
15-Sep-1984 23:37:58
14-Sep-1984 11:52:23

VAX-11 Bliss-32 V4.0-742
[ACLEDT.SRC]AEDDECODE.B32;1

Page 29
(4)

; Routine Size: 18 bytes, Routine Base: \$CODE\$ + 0243

```
398 0846 1 ROUTINE SET_DEFINITION =
399 0847 1
400 0848 1 ++
401 0849 1
402 0850 1 FUNCTIONAL DESCRIPTION:
403 0851 1
404 0852 1 This routine replaces a default definition with one from the
405 0853 1 action definition file.
406 0854 1
407 0855 1 CALLING SEQUENCE:
408 0856 1 SET_DEFINITION ()
409 0857 1
410 0858 1 INPUT PARAMETERS:
411 0859 1 none
412 0860 1
413 0861 1 IMPLICIT INPUTS:
414 0862 1 KEY_ACTION: ACL editor action code
415 0863 1 KEY_FLAGS: flags associated with the key definition
416 0864 1 KEY_STRING: descriptor of the string that defines a key
417 0865 1
418 0866 1 OUTPUT PARAMETERS:
419 0867 1 none
420 0868 1
421 0869 1 IMPLICIT OUTPUTS:
422 0870 1 none
423 0871 1
424 0872 1 ROUTINE VALUE:
425 0873 1 1
426 0874 1
427 0875 1 SIDE EFFECTS:
428 0876 1 The definition table is updated to reflect the new key definition.
429 0877 1
430 0878 1 --
431 0879 1
432 0880 2 BEGIN
433 0881 2
434 0882 2 LITERAL
435 0883 2 CHAR_CSI = 'X'9B', | C1 CSI character
436 0884 2 CHAR_CSI_1 = 'X'1B', | C0 CSI
437 0885 2 CHAR_CSI_2 = 'X'5B', | equivalent
438 0886 2 CHAR_SS3 = 'X'8F', | C1 SS3 character
439 0887 2 CHAR_SS3_1 = 'X'1B', | C0 SS3
440 0888 2 CHAR_SS3_2 = 'X'4F', | equivalent
441 0889 2
442 0890 2 LOCAL
443 0891 2 LOCAL STATUS, | Local error status
444 0892 2 NEW_KEY : REF $BLOCK, | Address of new definition storage
445 0893 2 NEXT_DEF : REF $BLOCK, | Address of next key definition
446 0894 2 KEY_INSERTED, | Flag to indicate key inserted
447 0895 2 TERM_OFFSET; | Size of overhead sequence
448 0896 2
449 0897 2 ! Check for angle bracket delimiters. If present, there must be a matched pair.
450 0898 2
451 0899 2 IF .KEY_STRING[DSC$W_LENGTH] GTR 1
452 0900 2 THEN
453 0901 2 BEGIN
454 0902 2 IF .VECTOR[.KEY_STRING[DSC$A_POINTER], 0; .BYTE] EQL '<'
```



```
455 0903 3 THEN
456 0904 4 BEGIN
457 0905 5 KEY_STRING[DSCSA_POINTER] = .KEY_STRING[DSCSA_POINTER] + 1;
458 0906 6 KEY_STRING[DSCSW_LENGTH] = .KEY_STRING[DSCSW_LENGTH] - 2;
459 0907 7 IF .VECTOR[.KEY_STRING[DSCSA_POINTER], .KEY_STRING[DSCSW_LENGTH]; .BYTE] NEQ '>'
460 0908 8 THEN RETURN 0;
461 0909 9 END;
462 0910 10
463 0911 11
464 0912 12 ! Check for conflicting type definitions.
465 0913 13
466 0914 14 IF (.KEY_BLOCK[KEY_V_CTRLCHAR] AND .KEY_BLOCK[KEY_V_ESCSEQ])
467 0915 15 OR (.KEY_BLOCK[KEY_V_CTRLCHAR] AND .KEY_STRING[DSCSW_LENGTH] NEQ 1)
468 0916 16 THEN RETURN 0;
469 0917 17
470 0918 18 ! If this is a C1 type definition, loop twice (once for the C1 definition
471 0919 19 ! and once for the C0 equivalent definition). Otherwise, only go through
472 0920 20 ! once.
473 0921 21
474 0922 22 INCR J FROM 1 TO (IF .KEY_BLOCK[KEY_V_CSI] OR .KEY_BLOCK[KEY_V_SS3]
475 0923 23 THEN 2 ELSE 1)
476 0924 24 DO
477 0925 25 BEGIN
478 0926 26
479 0927 27 ! Determine the size of the overhead area.
480 0928 28
481 0929 29 TERM_OFFSET = (IF .KEY_BLOCK[KEY_V_CSI] OR .KEY_BLOCK[KEY_V_SS3]
482 0930 30 THEN J
483 0931 31 ELSE IF .KEY_BLOCK[KEY_V_ESCSEQ]
484 0932 32 THEN 1
485 0933 33 ELSE 0);
486 0934 34
487 0935 35 ! Allocate storage for the key definition block.
488 0936 36
489 0937 37 AED_L_WORSTERR = ALLOCATE (.KEY_STRING[DSCSW_LENGTH] + KEY_C_LENGTH +
490 0938 38 + .TERM_OFFSET, NEW_KEY);
491 0939 39 IF NOT .AED_L_WORSTERR THEN RETURN 0;
492 0940 40
493 0941 41 ! Save the needed information in the key definition block.
494 0942 42
495 0943 43 NEW_KEY[KEY_B_ACTION] = .KEY_ACTION;
496 0944 44 NEW_KEY[KEY_B_SIZE] = .KEY_STRING[DSCSW_LENGTH] + .TERM_OFFSET;
497 0945 45 NEW_KEY[KEY_B_FLAGS] = .KEY_FLAGS OR KEY_M_USERDEF;
498 0946 46
499 0947 47 ! Set up the overhead area for the key text definition.
500 0948 48
501 0949 49 IF .KEY_BLOCK[KEY_V_CSI] OR .KEY_BLOCK[KEY_V_SS3]
502 0950 50 THEN
503 0951 51 BEGIN
504 0952 52 IF .J EQL 1
505 0953 53 THEN NEW_KEY[KEY_T_TEXT] = (IF .KEY_BLOCK[KEY_V_CSI]
506 0954 54 THEN CHAR_CSI ELSE CHAR_SS3)
507 0955 55 ELSE
508 0956 56 BEGIN
509 0957 57 NEW_KEY[KEY_T_TEXT] = (IF .KEY_BLOCK[KEY_V_CSI]
510 0958 58 THEN CHAR_CSI 1 ELSE CHAR_SS3 1);
511 0959 59 (NEW_KEY[KEY_T_TEXT]) + 1 = (IF .KEY_BLOCK[KEY_V_CSI]
```

```
512 0960 5
513 0961 5
514 0962 5
515 0963 5
516 0964 5
517 0965 5
518 0966 5
519 0967 5
520 0968 5
521 0969 5
522 0970 5
523 0971 5
524 0972 5
525 0973 5
526 0974 5
527 0975 5
528 0976 5
529 0977 5
530 0978 5
531 0979 5
532 0980 5
533 0981 5
534 0982 5
535 0983 5
536 0984 5
537 0985 5
538 0986 5
539 0987 5
540 0988 5
541 0989 5
542 0990 5
543 0991 5
544 0992 5
545 0993 5
546 0994 5
547 0995 5
548 0996 5
549 0997 5
550 0998 5
551 0999 5
552 1000 5
553 1001 5
554 1002 5
555 1003 5
556 1004 5

END;
THEN CHAR_CSI_2 ELSE CHAR_SS3_2);

ELSE IF .KEY_BLOCK[KEY_V_ESCSEQ]
THEN NEW_KEY[KEY_T_TEXT] = 'X'1B'
ELSE IF .KEY_BLOCK[KEY_V_CTRLCHAR]
THEN .KEY_STRING[DSCSA_POINTER] = ..KEY_STRING[DSCSA_POINTER] - 'X'40';

! Move over the key definition text.
CHSMOVE (.KEY_STRING[DSCSW_LENGTH], .KEY_STRING[DSCSA_POINTER],
NEW_KEY[KEY_T_TEXT] + .TERM_OFFSET);

! Check for and remove any default definitions that this new definition
! replaces.
NEXT_DEF = .KEY_TABLE[KEY_L_FLINK];
KEY_INSERTED = 0;
UNTIL .NEXT_DEF EQLA KEY_TABLE[KEY_L_FLINK]
DO
BEGIN
IF .NEXT_DEF[KEY_B_ACTION] EQL .KEY_ACTION
THEN
BEGIN
IF .KEY_INSERTED EQL 0
THEN
BEGIN
INSQUE (NEW_KEY[KEY_L_FLINK], NEXT_DEF[KEY_L_FLINK]);
KEY_INSERTED = 1;
END;
IF NOT .NEXT_DEF[KEY_V_USERDEF]
THEN
BEGIN
NEW_KEY = .NEXT_DEF[KEY_L_FLINK];
REMOVE (NEXT_DEF[KEY_L_FLINK], KEY_INSERTED);
NEXT_DEF = .NEW_KEY;
END;
END;
NEXT_DEF = .NEXT_DEF[KEY_L_FLINK];
END;
! End of C1 loop

END;
KEY_FLAGS = 0;
RETURN 1;

! End of routine SET_DEFINITION
```

OFFC 00000 SET_DEFINITION:

5E		08	C2	00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	0846
01	0000'	CF	B1	00005	SUBL2	#8, SP	0899
		1F	1B	0000A	CMPW	KEY_STRING, #1	
3C	0000'	DF	91	0000C	BLEQU	1\$	0902
		18	12	00011	CMPB	@KEY_STRING+4, #6U	
					BNEQ	1\$	

			0000'	CF	0000'	CF	D6	00013	INCL	KEY_STRING+4	0905	
				50		02	A2	00017	SUBW2	#2, KEY_STRING	0906	
				50	0000'	CF	3C	0001C	MOVZWL	KEY_STRING, R0	0907	
				3E	0000'	CF	C0	00021	ADDL2	KEY_STRING+4, R0		
						60	91	00026	CMPB	(R0), #62		
						0C	12	00029	BNEQ	28		
16		0000'	CF			03	E1	0002B	BBC	#3, KEY_BLOCK+10, 48	0914	
03		0000'	CF			04	E1	00031	BBC	#4, KEY_BLOCK+10, 38		
					0142	31	00037	28:	BRW	278		
07		0000'	CF			03	E1	0003A	38:	BBC	#3, KEY_BLOCK+10, 48	0915
		01		0000'	CF	B1	00040		CMPW	KEY_STRING, #1		
					F0	12	00045		BNEQ	28		
05		0000'	06		0000'	CF	E8	00047	48:	BLBS	KEY_BLOCK+10, 58	0922
		CF				01	E1	0004C		BBC	#1, KEY_BLOCK+10, 68	
		5B				02	D0	00052	58:	MOVL	#2, R11	
						03	11	00055		BRB	78	
		5B				01	D0	00057	68:	MOVL	#1, R11	
						59	D4	0005A	78:	CLRL	J	
					010F	31	0005C		BRW	268		
05		0000'	06		0000'	CF	E8	0005F	88:	BLBS	KEY_BLOCK+10, 98	0929
		CF				01	E1	00064		BBC	#1, KEY_BLOCK+10, 108	
		57				59	D0	0006A	98:	MOVL	J, TERM_OFFSET	0930
						0D	11	0006D		BRB	128	
05		0000'	CF			04	E1	0006F	108:	BBC	#4, KEY_BLOCK+10, 118	0931
		57				01	D0	00075		MOVL	#1, TERM_OFFSET	
						02	11	00078		BRB	128	
						57	D4	0007A	118:	CLRL	TERM_OFFSET	
					04	AE	9F	0007C	128:	PUSHAB	NEW_KEY	0938
		50			0000'	CF	3C	0007F		MOVZWL	KEY_STRING, R0	
	04	AE			0B	A740	9E	00084		MOVAB	11(TERM_OFFSET)[R0], 4(SP)	
					04	AE	9F	0008A		PUSHAB	4(SP)	
	00000000G	00				02	FB	0008D		CALLS	#2, LIB\$GET_VM	
		58				50	D0	00094		MOVL	R0, VM_STATUS	
		11				58	E9	00097		BLBC	VM_STATUS, 138	
		50			0000'	CF	3C	0009A		MOVZWL	KEY_STRING, R0	
		50			0B	A740	9E	0009F		MOVAB	11(TERM_OFFSET)[R0], R0	
50		6E				00	2C	000A4		MOVCS	#0, (SPT), #0, R0, @NEW_KEY	
					04	BE		000A9				
		0000'	CF			58	D0	000AB	138:	MOVL	VM_STATUS, AED_L_WORSTERR	
		82			0000'	CF	E9	000B0		BLBC	AED_L_WORSTERR, 28	0939
		50			04	AE	D0	000B5		MOVL	NEW_KEY, R0	0943
		0B			0000'	CF	90	000B9		MOVB	KEY_ACTION, 8(R0)	
	09	A0	0000'	CF		57	81	000BF		ADDB3	TERM_OFFSET, KEY_STRING, 9(R0)	0944
	0A	A0	0000'	CF		20	89	000C6		BISB3	#32, KEY_FLAGS, 10(R0)	0945
52	0000'	CF				00	EF	000CD		EXTZV	#0, #1, KEY_BLOCK+10, R2	0949
						52	E8	000D4		BLBS	R2, 148	
	32	0000'	CF			01	E1	000D7		BBC	#1, KEY_BLOCK+10, 208	
		01				59	D1	000DD	148:	CMPB	J, #1	0952
						13	12	000E0		BNEQ	178	
		06				52	E9	000E2		BLBC	R2, 158	0953
		51			9B	8F	9A	000E5		MOVZBL	#155, R1	
						04	11	000E9		BRB	168	
		51			8F	8F	9A	000EB	158:	MOVZBL	#143, R1	
	0B	A0				51	90	000EF	168:	MOVB	R1, 11(R0)	
						35	11	000F3		BRB	228	
		51				1B	D0	000F5	178:	MOVL	#27, R1	0957
	0B	A0				51	90	000F8		MOVB	R1, 11(R0)	

		06		52	E9	000FC	BLBC	R2, 188	0959
		51	5B	8F	9A	000FF	MOVZBL	#91, R1	
				04	11	00103	BRB	198	
		51	4F	8F	9A	00105	MOVZBL	#79, R1	
	0C	A0		51	D0	00109	MOVL	R1, 12(R0)	
				1B	11	0010D	BRB	228	0949
06	0000'	CF		04	E1	0010F	BBC	#4, KEY_BLOCK+10, 218	0963
	0B	A0		1B	90	00115	MOVB	#27, 11(R0)	0964
				0F	11	00119	BRB	228	
09	0000'	CF		03	E1	0011B	BBC	#3, KEY_BLOCK+10, 228	0965
	0000'	DF	00000040	8F	C2	00121	SUBL2	#64, @KEY_STRING+4	0966
0B A740	0000'	DF	0000'	CF	28	0012A	MOVCS	KEY_STRING, @KEY_STRING+4, 11(TERM_OFFSET)-[R0]	0971
		56	0000G	CF	D0	00134	MOVL	KEY_TABLE, NEXT_DEF	0976
				5A	D4	00139	CLRL	KEY_INSERTED	0977
		50	0000G	CF	9E	0013B	MOVAB	KEY_TABLE, R0	0978
		50		56	D1	00140	CMPL	NEXT_DEF, R0	
				29	13	00143	BEOL	268	
	0000'	CF	0B	A6	91	00145	CMPB	8(NEXT_DEF), KEY_ACTION	0981
				1C	12	0014B	BNEQ	258	
				5A	D5	0014D	TSTL	KEY_INSERTED	0984
				07	12	0014F	BNEQ	248	
		66	04	BE	0E	00151	INSQUE	@NEW KEY, (NEXT_DEF)	0987
		5A		01	D0	00155	MOVL	#1, KEY_INSERTED	0988
0C	0A	A6		05	E0	00158	BBS	#5, 10(NEXT_DEF), 258	0990
	04	AE	04	A6	D0	0015D	MOVL	4(NEXT_DEF), NEW KEY	0993
		5A		66	0F	00162	REMQUE	(NEXT_DEF), KEY_INSERTED	0994
		56	04	AE	D0	00165	MOVL	NEW KEY, NEXT_DEF	0995
		56		66	D0	00169	MOVL	(NEXT_DEF), NEXT_DEF	0998
				CD	11	0016C	BRB	238	0978
FEEB	59	01		5B	F1	0016E	ACBL	R11, #1, J. 88	0922
			0000'	CF	94	00174	CLRB	KEY_FLAGS	1001
		50		01	D0	00178	MOVL	#1, R0	1002
					04	0017B	RET		
				50	D4	0017C	CLRL	R0	1004
				04	0017E	RET			

; Routine Size: 383 bytes, Routine Base: \$CODE\$ + 0255


```
558 1005 1 GLOBAL ROUTINE AED_DECODEKEY =
559 1006 1
560 1007 1 **
561 1008 1
562 1009 1 FUNCTIONAL DESCRIPTION:
563 1010 1
564 1011 1 This routine accepts input from the input channel and decodes it
565 1012 1 according to the definitions from the action definition file (or
566 1013 1 the default definitions).
567 1014 1
568 1015 1 CALLING SEQUENCE:
569 1016 1 AED_DECODEKEY ()
570 1017 1
571 1018 1 INPUT PARAMETERS:
572 1019 1 none
573 1020 1
574 1021 1 IMPLICIT INPUTS:
575 1022 1 none
576 1023 1
577 1024 1 OUTPUT PARAMETERS:
578 1025 1 none
579 1026 1
580 1027 1 IMPLICIT OUTPUTS:
581 1028 1 none
582 1029 1
583 1030 1 ROUTINE VALUE:
584 1031 1 0 if a fatal error occurs,
585 1032 1 Action code value if special (AED_V_ACTION also set)
586 1033 1 ASCII character value
587 1034 1
588 1035 1 SIDE EFFECTS:
589 1036 1 none
590 1037 1
591 1038 1 --
592 1039 1
593 1040 2 BEGIN
594 1041 2
595 1042 2 MACRO
596 1043 2 TERM_CHAR = AED_W_IOSB[2] %;
597 1044 2 TERM_SIZE = AED_W_IOSB[3] %;
598 1045 2 TERM_STRING = INPUT_BUFFER[AED_W_IOSB[1]] %;
599 1046 2
600 1047 2 LABEL
601 1048 2 DECODE_KEY;
602 1049 2
603 1050 2 LOCAL
604 1051 2 LOCAL_STATUS, ! Local routine return status
605 1052 2 INPUT_BUFFER : VECTOR [10, BYTE], ! Storage for input characters
606 1053 2 TERM_DESC : $BBLOCK [DSC&C_S_BLN], ! Term table descr
607 1054 2 TERM_TABLE : VECTOR [8], ! Terminator table
608 1055 2 ! INITIAL (REP 8 OF (-1)), ! All are terminators
609 1056 2 NEXT_DEF : REF $BBLOCK, ! Address of next key definition
610 1057 2 KEY_WITHOUT_GLD : REF $BBLOCK, ! Address of key definition without gold required
611 1058 2 RETURN_CHAR; ! Character/code to return
612 1059 2
613 1060 2 ! If this is a recovery, get a word (16 bits) from the recovery record. This
614 1061 2 ! contains a character if the high byte is zero, or an editor action if not.
```



```
672 1119      END;
673 1120
674 1121      ! If the character is nothing special, simply return with the character.
675 1122
676 1123      AED_L_FLAGS[AED_V_ACTIONKEY] = 0;
677 1124      IF .TERM_CHAR GEQ ' ' AND .TERM_CHAR NEQ 'X'7F'
678 1125      THEN
679 1126          BEGIN
680 1127              RETURN_CHAR = .TERM_CHAR;
681 1128              LEAVE DECODE_KEY;
682 1129          END;
683 1130
684 1131      ! Otherwise, it will be necessary to search the action definition table to
685 1132      ! determine whether or not the character (or characters) defines an ACL
686 1133      ! editor action.
687 1134
688 1135      KEY_WITHOUT_GLD = 0;
689 1136      NEXT_DEF = .KEY_TABLE[KEY_L_FLINK];
690 1137      UNTIL .NEXT_DEF EQL KEY_TABLE[KEY_L_FLINK]
691 1138      DO
692 1139          BEGIN
693 1140              IF CH$EQL (.NEXT_DEF[KEY_B_SIZE], NEXT_DEF[KEY_T_TEXT],
694 1141                      .TERM_SIZE, TERM_STRING, 0)
695 1142              THEN
696 1143                  BEGIN
697 1144                      IF .NEXT_DEF[KEY_V_GOLDREQ] EQL .AED_L_FLAGS[AED_V_GOLDKEY]
698 1145                      THEN
699 1146                          BEGIN
700 1147                              AED_L_FLAGS[AED_V_ACTIONKEY] = 1;
701 1148                              RETURN_CHAR = .NEXT_DEF[KEY_B_ACTION];
702 1149                              LEAVE DECODE_KEY;
703 1150                          END;
704 1151                      IF NOT .NEXT_DEF[KEY_V_GOLDREQ] THEN KEY_WITHOUT_GLD = .NEXT_DEF;
705 1152                      END;
706 1153                      NEXT_DEF = .NEXT_DEF[KEY_L_FLINK];
707 1154                  END;
708 1155
709 1156      ! Nothing has been found in the definition table. Check to see if there
710 1157      ! was a key defined except that the gold key was hit but not required.
711 1158      ! If this is the case, clear the GOLDKEY flag and return the appropriate
712 1159      ! action code. Otherwise simply return the terminating character.
713 1160
714 1161      IF .KEY_WITHOUT_GLD NEQ 0
715 1162      THEN
716 1163          BEGIN
717 1164              AED_L_FLAGS[AED_V_GOLDKEY] = 0;
718 1165              AED_L_FLAGS[AED_V_ACTIONKEY] = 1;
719 1166              RETURN_CHAR = .KEY_WITHOUT_GLD[KEY_B_ACTION];
720 1167              LEAVE DECODE_KEY;
721 1168          END;
722 1169      RETURN_CHAR = .TERM_CHAR;
723 1170      END;                                     ! End of DECODE_KEY block
724 1171
725 1172      ! If the action cannot be logged (EXIT or QUIT), simply return now.
726 1173
727 1174      IF .AED_L_FLAGS[AED_V_ACTIONKEY]
728 1175      AND (.RETURN_CHAR EQL KEY_C_EXIT OR .RETURN_CHAR EQL KEY_C_QUIT)
```

```
! End of routine AED_DECODEKEY
```

Address	Hex	Label	Instruction	Comment	Offset
00000000	00	9E	00002		
00000000	5B	00000000G	00	9E	00002
00000000	5A	0000	CF	9E	00009
00000000	5E		34	C2	0000E
00000000	CF		20	28	00011
00000000	AA		01	E1	00017
00000000		04F6	CA	B5	0001C
00000000			46	12	00020
00000000		04D4	CA	9F	00022
00000000G	00		01	FB	00026
0001827A	34		50	E8	00020
	8F		50	D1	00030
			18	13	00037
	7E	04DC	CA	7D	00039
		0424	CA	9F	0003E
		00000000G	8F	DD	00042
0000G	CF		04	FB	00048

	04	AA		02	8A	0004D	BICB2	#2, AED_B_OPTIONS	1077
			0424	CA	9F	00051	PUSHAB	RECOVER_FAB	1079
	00000000G	00		01	FB	00055	CALLS	#1, SYSSCLOSE	
	04	AA		02	8A	0005C	BICB2	#2, AED_B_OPTIONS	1080
		50		01	D0	00060	MOVL	#1, R0	1081
				04	00063		RET		
			0524	CA	D4	00064	CLRL	RECOVER_INDEX	1083
		50	0518	CA	9E	00068	MOVAB	RECOVER_BUFFER, R0	1085
		57	0524	DA40	9A	0006D	MOVZBL	@RECOVER_INDEX[R0], RETURN_CHAR	
			0524	CA	D6	00073	INCL	RECOVER_INDEX	1086
		50	0518	CA	9E	00077	MOVAB	RECOVER_BUFFER, R0	1087
			0524	DA40	9F	0007C	PUSHAB	@RECOVER_INDEX[R0]	
02	AA	01		9E	F0	00081	INSV	@(SP)+, #5, #1, AED_L_FLAGS+2	
			0524	CA	D6	00087	INCL	RECOVER_INDEX	1088
	04F6	CA		02	A2	0008B	SUBW2	#2, RECOVER_RAB+34	1089
				50	11	00090	BRB	6\$	1063
	20	AE		20	B0	00092	MOVW	#32, TERM_DESC	1096
	24	AE		6E	9E	00096	MOVAB	TERM_TABLE, TERM_DESC+4	1097
				7E	7C	0009A	CLRQ	-(SP)	1105
		7E	28	AE	9F	0009C	PUSHAB	TERM_DESC	
				0A	7D	0009F	MOVQ	#10, -(SP)	
			3C	AE	9F	000A2	PUSHAB	INPUT_BUFFER	
				7E	7C	000A5	CLRQ	-(SP)	
		7E	0084	CA	9F	000A7	PUSHAB	AED_W_IOSB	
		7E	5231	8F	3C	000AB	MOVZWL	#2104T, -(SP)	
			7C	AA	3C	000B0	MOVZWL	AED_W_TERMIN, -(SP)	
				7E	D4	000B4	CLRL	-(SP)	
	00000000G	00		0C	FB	000B6	CALLS	#12, SYSSQIOW	
	008C	CA		50	D0	000BD	MOVL	R0, AED_L_STATUS	
		0C	008C	CA	E9	000C2	BLBC	AED_L_STATUS, 5\$	1106
	008C	CA	0084	CA	3C	000C7	MOVZWL	AED_W_IOSB, AED_L_STATUS	
		60	008C	CA	E8	000CE	BLBS	AED_L_STATUS, 1T\$	1107
		3C	008C	CA	D1	000D3	CMPL	AED_L_STATUS, #60	1110
				0B	12	000D8	BNEQ	7\$	
	008C	CA		01	D0	000DA	MOVL	#1, AED_L_STATUS	1113
		57		1B	D0	000DF	MOVL	#27, RETURN_CHAR	1114
			00C7	31	000E2		BRW	18\$	1115
				03	E1	000E5	BBC	#3, AED_L_FLAGS, 8\$	1117
		6A		01	DD	000E9	PUSHL	#1	
				15	DD	000EB	PUSHL	#21	
	00000000G	00		02	FB	000ED	CALLS	#2, SCR\$ERASE_PAGE	
				01	DD	000F4	PUSHL	#1	
				15	DD	000F6	PUSHL	#21	
		6B		02	FB	000F8	CALLS	#2, SCR\$SET_CURSOR	
			008C	CA	DD	000FB	PUSHL	AED_L_STATUS	8\$:
	00000000G	00		01	FB	000FF	CALLS	#1, LIB\$SIGNAL	
0B		6A		03	E1	00106	BBC	#3, AED_L_FLAGS, 9\$	
		7E	20	AA	9A	0010A	MOVZBL	AED_B_COLUMN, -(SP)	
		7E	24	AA	9A	0010E	MOVZBL	AED_B_LINE, -(SP)	
		6B		02	FB	00112	CALLS	#2, SCR\$SET_CURSOR	
		50	008C	CA	D0	00115	MOVL	AED_L_STATUS, R0	9\$:
		07		50	93	0011A	BITB	R0, #7	
				11	13	0011D	BEQL	10\$	
51		03		00	EF	0011F	EXTZV	#0, #3, R0, R1	
51	14	50		00	ED	00124	CMPZV	#0, #3, AED_L_WORSTERR, R1	
		AA		04	18	0012A	BGEQ	10\$	
				50	D0	0012C	MOVL	R0, AED_L_WORSTERR	
	14	AA							

				00DE	31	00130	10\$:	BRW	25\$		1118
	02	AA		20	8A	00133	11\$:	BICB2	#32, AED_L_FLAGS+2		1123
		58		0088	CA	3C		MOVZWL	AED_W_IOSB+4, R8		1124
		20			58	B1		CMPW	R8, #32		
					07	1F		BLSSU	12\$		
	007F	8F			58	B1		CMPW	R8, #127		
					61	12		BNEQ	17\$		
					56	D4	12\$:	CLRL	KEY_WITHOUT_GLD		1135
		54		0000G	CF	D0		MOVL	KEY_TABLE, NEXT_DEF		1136
		55		0086	CA	3C		MOVZWL	AED_W_IOSB+2, R5		1141
		59		28	AE	9E		MOVAB	INPUT_BUFFER, R9		
		50		0000G	CF	9E	13\$:	MOVAB	KEY_TABLE, R0		1137
		50			54	D1		CMP	NEXT_DEF, R0		
					35	13		BEQL	16\$		
					50	9A		MOVZBL	9(NEXT_DEF), R0		1140
008A	CA		00	08	A4	2D		CMP	R0, 11(NEXT_DEF), #0, AED_W_IOSB+6, (R9)-		1141
					6945			[R5]			
					20	12		BNEQ	15\$		
					03	EF		EXTZV	#3, #1, AED_L_FLAGS+1, R0		1144
					02	ED		CMPZV	#2, #1, 10(NEXT_DEF), R0		
					0A	12		BNEQ	14\$		
					20	88		BISB2	#32, AED_L_FLAGS+2		1147
					08	A4		MOVZBL	8(NEXT_DEF), RETURN_CHAR		1148
					22	11		BRB	18\$		1149
					02	E0	14\$:	BBS	#2, 10(NEXT_DEF), 15\$		1151
					54	D0		MOVL	NEXT_DEF, KEY_WITHOUT_GLD		
					64	D0	15\$:	MOVL	(NEXT_DEF), NEXT_DEF		1153
					C1	11		BRB	13\$		1137
					56	D5	16\$:	TSTL	KEY_WITHOUT_GLD		1161
					0E	13		BEQL	17\$		
					08	8A		BICB2	#8, AED_L_FLAGS+1		1164
					20	88		BISB2	#32, AED_L_FLAGS+2		1165
					08	A6		MOVZBL	8(KEY_WITHOUT_GLD), RETURN_CHAR		1166
					03	11		BRB	18\$		1167
					58	D0	17\$:	MOVL	R8, RETURN_CHAR		1169
					05	E1	18\$:	BBC	#5, AED_L_FLAGS+2, 19\$		1174
					57	D1		CMP	RETURN_CHAR, #39		1175
					57	13		BEQL	24\$		
					28	D1		CMP	RETURN_CHAR, #40		
					52	13		BEQL	24\$		
					04	AA	19\$:	BLBC	AED_B_OPTIONS, 24\$		1181
					0420	CA		CMP	JOURNAL_INDEX, #10		1184
					1E	19		BLSS	21\$		
					0378	CA		PUSHAB	JOURNAL_RAB		1187
					01	FB		CALLS	#1, SYSSPUT		
					50	E8		BLBS	R0, 20\$		
					01	8A		BICB2	#1, AED_B_OPTIONS		
					00	2C	20\$:	MOV	#0, (SPT), #0, #10, JOURNAL_BUFFER		1188
					0414	CA					
					0420	CA		CLRL	JOURNAL_INDEX		1189
					0414	CA	21\$:	MOVAB	JOURNAL_BUFFER, R0		1191
					57	90		MOV	RETURN_CHAR, @JOURNAL_INDEX[R0]		
					0420	CA		INCL	JOURNAL_INDEX		1192
					0414	CA		MOVAB	JOURNAL_BUFFER, R0		1194
					0420	CA		ADDL2	JOURNAL_INDEX, R0		
					05	E1		BBC	#5, AED_L_FLAGS+2, 22\$		1193
					01	90		MOV	#1, (R0)		1194

AED\$DECODE
V04-000

0 3
15-Sep-1984 23:37:58
14-Sep-1984 11:52:23

VAX-11 Bliss-32 V4.0-742
[ACLEDT.SRC]AEDDECODE.B32;1

Page 41
(6)

	02	11	00205		BRB	23\$	
	60	94	00207	22\$:	CLRB	(R0)	
	CA	D6	00209	23\$:	INCL	JOURNAL_INDEX	
50	57	D0	0020D	24\$:	MOVL	RETURN_CHAR, R0	
		04	00210		RET		
	50	D4	00211	25\$:	CLRL	R0	
		04	00213		RET		

: 1195
: 1196
: 1199
: 1201
:

; Routine Size: 532 bytes, Routine Base: \$CODE\$ + 03D4

```
1202 1 GLOBAL ROUTINE AED_FLUSHKEY =
1203 1
1204 1 ++
1205 1
1206 1 FUNCTIONAL DESCRIPTION:
1207 1
1208 1     This routine flushes the journal buffer and closes the journal file.
1209 1
1210 1 CALLING SEQUENCE:
1211 1     AED_FLUSHKEY ()
1212 1
1213 1 INPUT PARAMETERS:
1214 1     none
1215 1
1216 1 IMPLICIT INPUTS:
1217 1     OWN storage
1218 1
1219 1 OUTPUT PARAMETERS:
1220 1     none
1221 1
1222 1 IMPLICIT OUTPUTS:
1223 1     none
1224 1
1225 1 ROUTINE VALUE:
1226 1     1
1227 1
1228 1 SIDE EFFECTS:
1229 1     none
1230 1
1231 1 --
1232 1
1233 2 BEGIN
1234 2
1235 2 ! If not writing a journal file, simply return now.
1236 2
1237 2 IF NOT .AED_B_OPTIONS[AED_V_JOURNAL] THEN RETURN 1;
1238 2
1239 2 IF .JOURNAL_INDEX GTR 0
1240 2 THEN
1241 2     BEGIN
1242 2         JOURNAL_RAB[RAB$W_RSZ] = .JOURNAL_INDEX * 2;
1243 2         $PUT (RAB = JOURNAL_RAB);
1244 2     END;
1245 2
1246 2 JOURNAL_FAB[FAB$V_DLT] = NOT .AED_B_OPTIONS[AED_V_KEEPJNL];
1247 2 $CLOSE (FAB = JOURNAL_FAB);
1248 2
1249 2 RETURN 1;
1250 2
1251 1 END;

! End of routine AED_FLUSHKEY
```

```
34      0000'  CF  0000 00000
          E9 00002
```

```
.ENTRY  AED_FLUSHKEY, Save nothing
BLBC   AED_B_OPTIONS, 2$
```

```
: 1202
: 1237
```


AED\$DECODE
V04-000

F 3
15-Sep-1984 23:37:58
14-Sep-1984 11:52:23

VAX-11 Bliss-32 V4.0-742
[ACLEDT.SRC]AEDDECODE.B32;1

Page 43
(7)

			50	0000'	CF	D0	00007		MOVL	JOURNAL_INDEX, R0		1239
						11	15	0000C	BLEQ	1\$		
	0000'	CF	50			02	A5	0000E	MULW3	#2, R0, JOURNAL_RAB+34		1242
				0000'		CF	9F	00014	PUSHAB	JOURNAL_RAB		1243
			00	00000000G		01	FB	00018	CALLS	#1, SYS\$PUT		
50	0000'	CF	01			03	EF	0001F	EXTZV	#3, #1, AED_B_OPTIONS, R0		1246
			50			50	D2	00026	MCOML	R0, R0		
0000'	CF		07			50	F0	00029	INSV	R0, #7, #1, JOURNAL_FAB+5		
		01				50	F0	00029	PUSHAB	JOURNAL_FAB		1247
			00	00000000G		01	FB	00034	CALLS	#1, SYS\$CLOSE		
			50			01	D0	0003B	MOVL	#1, R0		1249
						04	0003E		RET			1251

: Routine Size: 63 bytes, Routine Base: \$CODE\$ + 05E8

: 806 1252 1
: 807 1253 1 END
: 808 1254 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes					
AED COMMON	1320	NOVEC, WRT, RD	, NOEXE, NOSHR,	LCL, REL,	OVR, NOPIC,	ALIGN(0)	
\$OWNS	20	NOVEC, WRT, RD	, NOEXE, NOSHR,	LCL, REL,	CON, NOPIC,	ALIGN(2)	
_LIB\$KEYOS	94	NOVEC, NOWRT, RD	, EXE, SHR,	LCL, REL,	CON, PIC,	ALIGN(1)	
_LIB\$STATES	538	NOVEC, NOWRT, RD	, EXE, SHR,	LCL, REL,	CON, PIC,	ALIGN(1)	
_LIB\$KEY1\$	518	NOVEC, NOWRT, RD	, EXE, SHR,	LCL, REL,	CON, PIC,	ALIGN(1)	
\$PLITS	52	NOVEC, NOWRT, RD	, NOEXE, NOSHR,	LCL, REL,	CON, NOPIC,	ALIGN(2)	
\$CODE\$	1575	NOVEC, NOWRT, RD	, EXE, NOSHR,	LCL, REL,	CON, NOPIC,	ALIGN(2)	

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	122	0	1000	00:01.8
_\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	29	69	14	00:00.2

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:AEDDECODE/OBJ=OBJ\$:AEDDECODE MSRC\$:AEDDECODE/UPDATE=(ENH\$:AEDDECODE)

AED\$DECODE
V04-000

6 3
15-Sep-1984 23:37:58
14-Sep-1984 11:52:23

VAX-11 Bliss-32 V4.0-742
[ACLEDT.SRC]AEDDECODE.B32:1

Page 44
(7)

: Size: 1575 code + 2542 data bytes
: Run Time: 01:10.7
: Elapsed Time: 03:37.1
: Lines/CPU Min: 1064
: Lexemes/CPU-Min: 71863
: Memory Used: 431 pages
: Compilation Complete

0002

**DIGITAL
CONFIDE**

0003 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY